

Actividad 6.1 (Seguimiento de Trayectorias)

Daniel Castillo López

A01737357

1. **Implementar** el código requerido para generar el seguimiento de las siguientes trayectorias con un robot tipo diferencial:

$$x = 2\cos(0.2*t), y = 2\sin(0.4*t),$$

```
%Limpieza de pantalla
clear all
close all
clc
```

En esta sección se definen los parámetros temporales de la simulación. Se establece un tiempo total de simulación y un intervalo de muestreo. A partir de estos valores se construye un vector de tiempo, que contiene todos los instantes en los que se evaluará la simulación. Finalmente, se calcula el número total de muestras N, que corresponde al número de elementos en el vector de tiempo, y que será usado para iterar a lo largo de la simulación.

```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=40;      % Tiempo de simulación en segundos (s)
ts=0.05;    % Tiempo de muestreo en segundos (s)
t=0:ts:tf;  % Vector de tiempo
N= length(t); % Muestras
```

Se especifican las condiciones iniciales del robot móvil. La posición inicial en los ejes x y y se fija en cero y su orientación (ángulo), como también se establece en cero, lo cual indica que el robot comienza en el origen con una orientación estándar. Además, se inicializan las variables hx y hy, que representan la posición del punto de control del robot, haciéndolas coincidir con las coordenadas iniciales del robot.

```
%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)
```

Aquí se define la trayectoria que se desea que el robot siga. Se utiliza una ecuación paramétrica para generar una trayectoria elíptica, dada por las expresiones, que definen las coordenadas x e y deseadas en función del tiempo. Luego, se calculan las velocidades deseadas como la derivada numérica de estas posiciones utilizando diferencias finitas.

```
%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=2*cos(0.2*t);
hyd=2*sin(0.4*t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[3.5 0;...
        0 7.5];

    %d)Velocidades deseadas
    hdp=[hxdp(k);hydp(k)];

    %e)Ley de Control:Agregamos las velocidades deseadas
    qpRef= pinv(J)*(hdp + K*he);

    v(k)= qpRef(1); %Velocidad lineal de entrada al robot
```

```
w(k)= qpRef(2);    %Velocidad angular de entrada al robot
```

Con las velocidades calculadas, se actualiza el estado del robot. Primero se integra la velocidad angular w para obtener el nuevo ángulo de orientación ϕ , utilizando el método de Euler. Luego, se calculan las componentes de velocidad lineal en los ejes x e y y se integran de igual forma para actualizar la posición del robot. Finalmente, se actualiza la posición del punto de control, que en este caso coincide con la posición del robot, reflejando el desplazamiento logrado durante ese intervalo de tiempo.

```
%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Aplico la integral a la velocidad angular para obtener el angulo "phi" de la orientación
```

```
phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
xp1=v(k)*cos(phi(k));
```

```
yp1=v(k)*sin(phi(k));
```

```
%Aplico la integral a la velocidad lineal para obtener las cordenadas
```

```
%"x1" y "y1" de la posición
```

```
x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
```

```
y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)
```

```
% Posicion del robot con respecto al punto de control
```

```
hx(k+1)=x1(k+1);
```

```
hy(k+1)=y1(k+1);
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% a) Configuracion de escena
```

```
scene=figure; % Crear figura (Escena)
```

```
set(scene,'Color','white'); % Color del fondo de la escena
```

```
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
```

```
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
```

```
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
```

```
camlight('headlight'); % Luz para la escena
```

```
axis equal; % Establece la relación de aspecto para que las unidades de datos sean las mismas en todas las direcciones.
```

```
grid on; % Mostrar líneas de cuadrícula en los ejes
```

```
box on; % Mostrar contorno de ejes
```

```
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje
```

```
view([-0.1 90]); % Orientacion de la figura
```

```

axis([-10 10 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

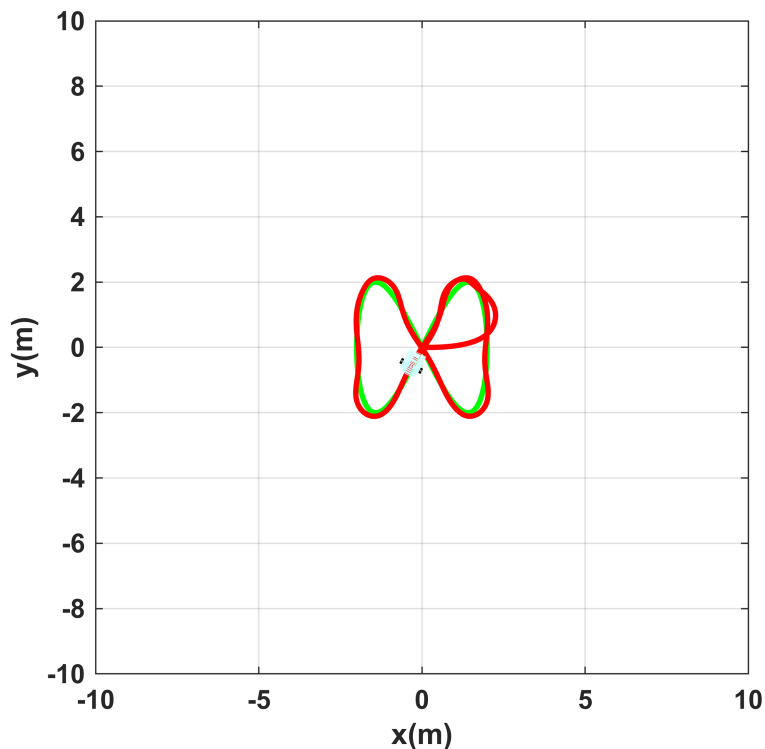
    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

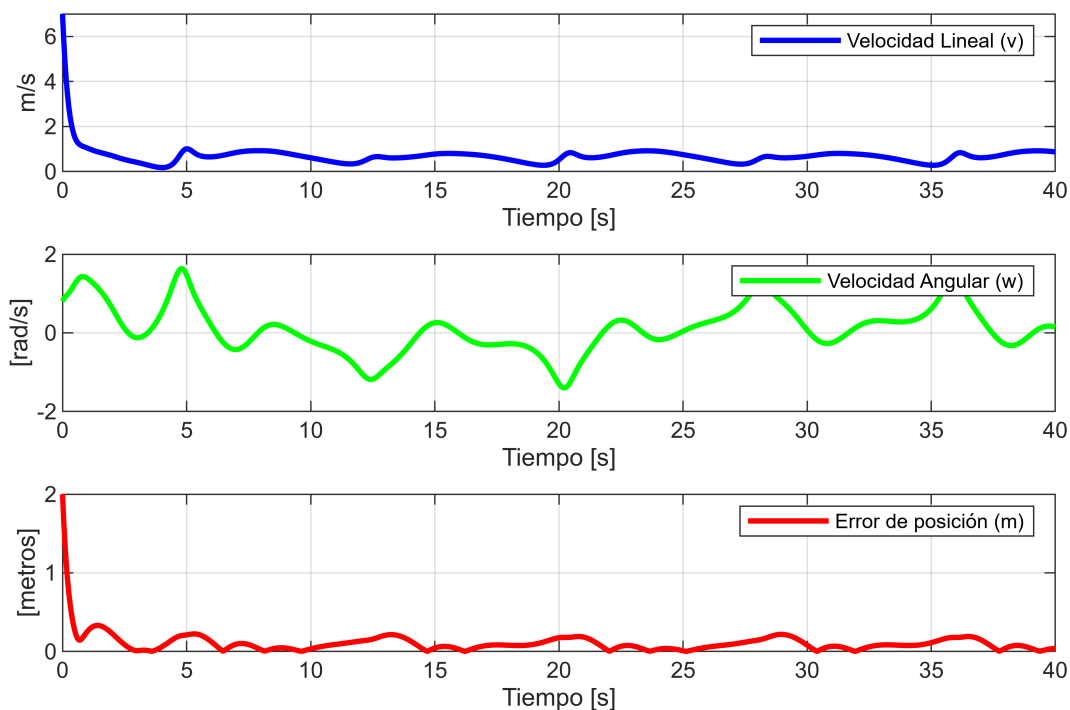
```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[metros]'),legend('Error de posición (m)');

```



b) $x = t - 3\sin(t)$, $y = 4 - 3\cos(t)$

```

%%
% Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf2=40;          % Tiempo de simulación en segundos (s)
ts2=0.05;        % Tiempo de muestreo en segundos (s)
t2=0:ts2:tf2;    % Vector de tiempo
N2= length(t2);  % Muestras

```

```

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x2(1)=0;      % Posición inicial eje x
y2(1)=0;      % Posición inicial eje y
phi2(1)=0;    % Orientación inicial del robot

hx2(1)= x2(1);      % Posición del punto de control eje X
hy2(1)= y2(1);      % Posición del punto de control eje Y

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hxd2 = t2 - 3*sin(t2);
hyd2 = 4 - 3*cos(t2);

hxdp2 = diff(hxd2)/ts2;
hydp2 = diff(hyd2)/ts2;

hxdp2(N2) = hxdp2(end);
hydp2(N2) = hydp2(end);

for k=1:N2
    hxe2(k) = hxd2(k) - hx2(k);
    hye2(k) = hyd2(k) - hy2(k);

    he2 = [hxe2(k); hye2(k)];
    Error2(k) = sqrt(hxe2(k)^2 + hye2(k)^2);

    J2 = [cos(phi2(k)) -sin(phi2(k)); sin(phi2(k)) cos(phi2(k))];

    K2 = [13 0; 0 13];

    hdp2 = [hxdp2(k); hydp2(k)];
    qpRef2 = pinv(J2)*(hdp2 + K2*he2);

    v2(k) = qpRef2(1);
    w2(k) = qpRef2(2);

    phi2(k+1) = phi2(k) + w2(k)*ts2;

    xp2 = v2(k)*cos(phi2(k));
    yp2 = v2(k)*sin(phi2(k));

    x2(k+1) = x2(k) + ts2*xp2;
    y2(k+1) = y2(k) + ts2*yp2;

    hx2(k+1) = x2(k+1);
    hy2(k+1) = y2(k+1);
end

% SIMULACION VIRTUAL 3D

```

```

scene2=figure;
set(scene2,'Color','white');
set(gca,'FontWeight','bold');
sizeScreen2=get(0,'ScreenSize');
set(scene2,'position',sizeScreen2);
camlight('headlight');
axis equal;
grid on;
box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([-0.1 90]);
axis([-5 20 -2 10 0 1]);

scale2 = 4;
MobileRobot_5;
H21=MobilePlot_4(x2(1),y2(1),phi2(1),scale2); hold on;
H22=plot3(hx2(1),hy2(1),0,'b','lineWidth',2);
H23=plot3(hxd2,hyd2,zeros(1,N2),'m','lineWidth',2);

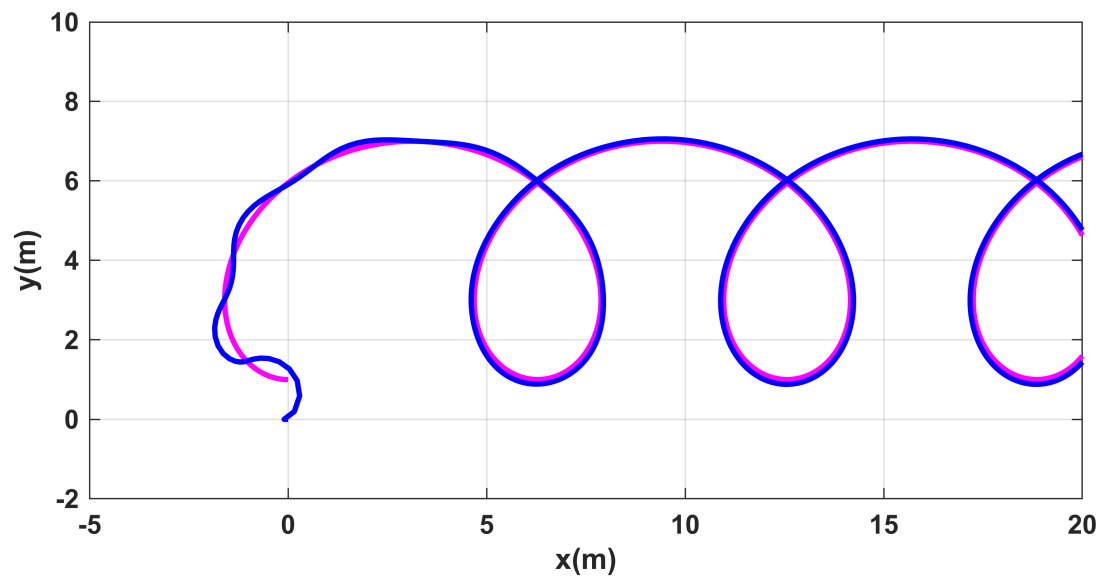
step2=1;

for k=1:step2:N2
    delete(H21);
    delete(H22);

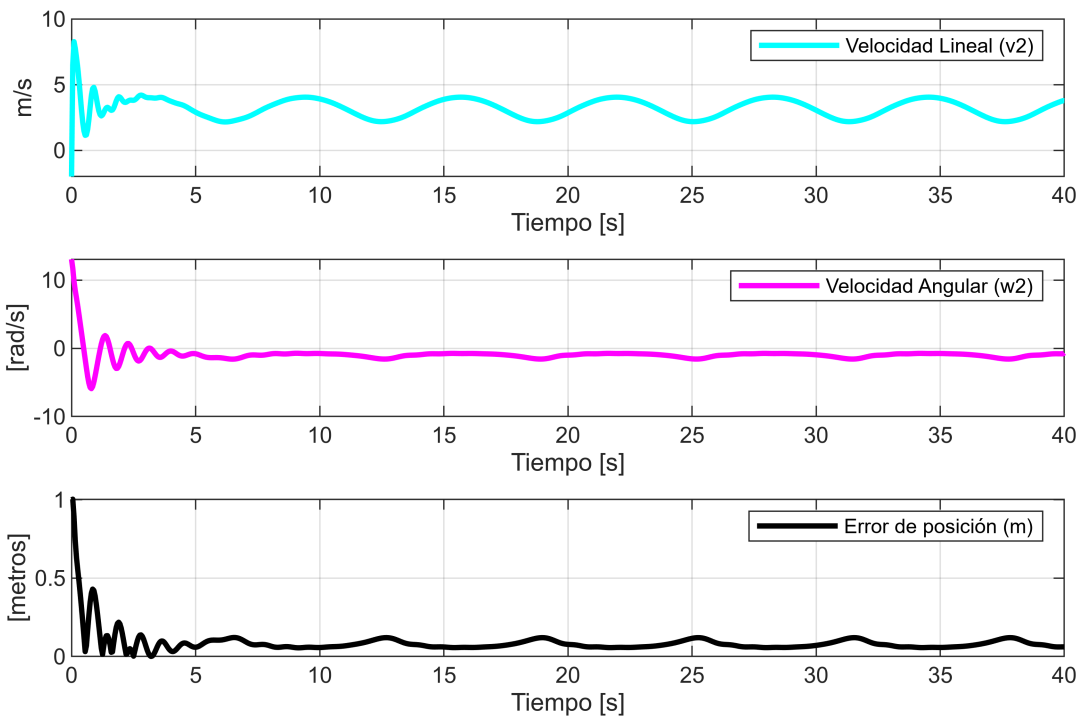
    H21=MobilePlot_4(x2(k),y2(k),phi2(k),scale2);
    H22=plot3(hx2(1:k),hy2(1:k),zeros(1,k),'b','lineWidth',2);

    pause(ts2);
end

```



```
% GRAFICAS
graph2=figure;
set(graph2,'position',sizeScreen2);
subplot(311)
plot(t2,v2,'c','LineWidth',2),grid on,xlabel('Tiempo [s]'),ylabel('m/
s'),legend('Velocidad Lineal (v2)');
subplot(312)
plot(t2,w2,'m','LineWidth',2),grid on,xlabel('Tiempo [s]'),ylabel('[rad/
s]'),legend('Velocidad Angular (w2)');
subplot(313)
plot(t2>Error2,'k','LineWidth',2),grid on,xlabel('Tiempo
[s]'),ylabel('[metros]'),legend('Error de posición (m)');
```

c) $x = 3\cos(t) - \cos(3t)$, $y = 4\sin(3t)$

```
%%
%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=7;          % Tiempo de simulación en segundos (s)
ts=0.01;       % Tiempo de muestreo en segundos (s)
t=0:ts:tf;     % Vector de tiempo
N= length(t);  % Muestras

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
```

```
hy(1)= y1(1);          % Posición del punto de control en el eje (Y) metros (m)
```

```
%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Ecuaciones paramétricas de la trayectoria deseada
```

```
%Figura 1
```

```
hxd=3*cos(t)-cos(3*t);
```

```
hyd=4*sin(3*t);
```

```
%Velocidades de la trayectoria deseada
```

```
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
```

```
hxdp = diff(hxd)/ts;
```

```
hydp = diff(hyd)/ts;
```

```
% Ajustamos tamaño para que coincidan con t
```

```
hxdp(N) = hxdp(N-1); % Último valor repetido
```

```
hydp(N) = hydp(N-1); % Último valor repetido
```

```
for k=1:N
```

```
    %a)Errores de control (Aqui la posición deseada ya no es constante,
```

```
    % varia con el tiempo)
```

```
    hxe(k)=hxd(k)-hx(k);
```

```
    hye(k)=hyd(k)-hy(k);
```

```
    %Matriz de error
```

```
    he= [hxe(k);hye(k)];
```

```
    %Magnitud del error de posición
```

```
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);
```

```
    %b)Matriz Jacobiana
```

```
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D  
      sin(phi(k)) cos(phi(k))];
```

```
    %c)Matriz de Ganancias
```

```
    K=[13 0;...  
      0 13];
```

```
    %d)Velocidades deseadas
```

```
    hdp=[hxdp(k);hydp(k)];
```

```
    %e)Ley de Control:Agregamos las velocidades deseadas
```

```
    qpRef= pinv(J)*(hdp + K*he);
```

```
    v(k)= qpRef(1); %Velocidad lineal de entrada al robot
```

```
    w(k)= qpRef(2); %Velocidad angular de entrada al robot
```

```
%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
    orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

view([-0.1 90]); % Orientacion de la figura
axis([-10 10 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada

```

```
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot
```

```
step=1; % pasos para simulacion
```

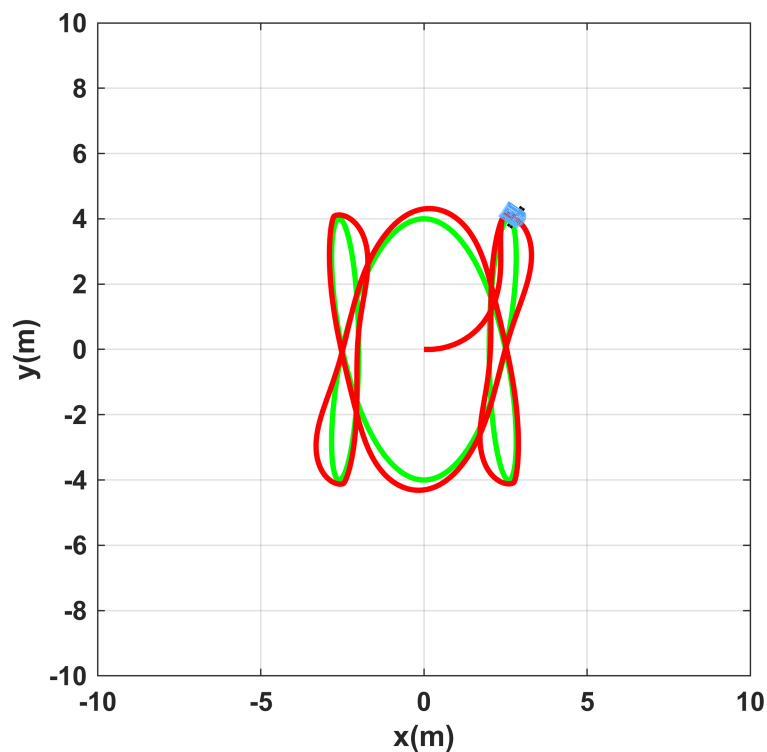
```
for k=1:step:N
```

```
    delete(H1);
    delete(H2);
```

```
    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);
```

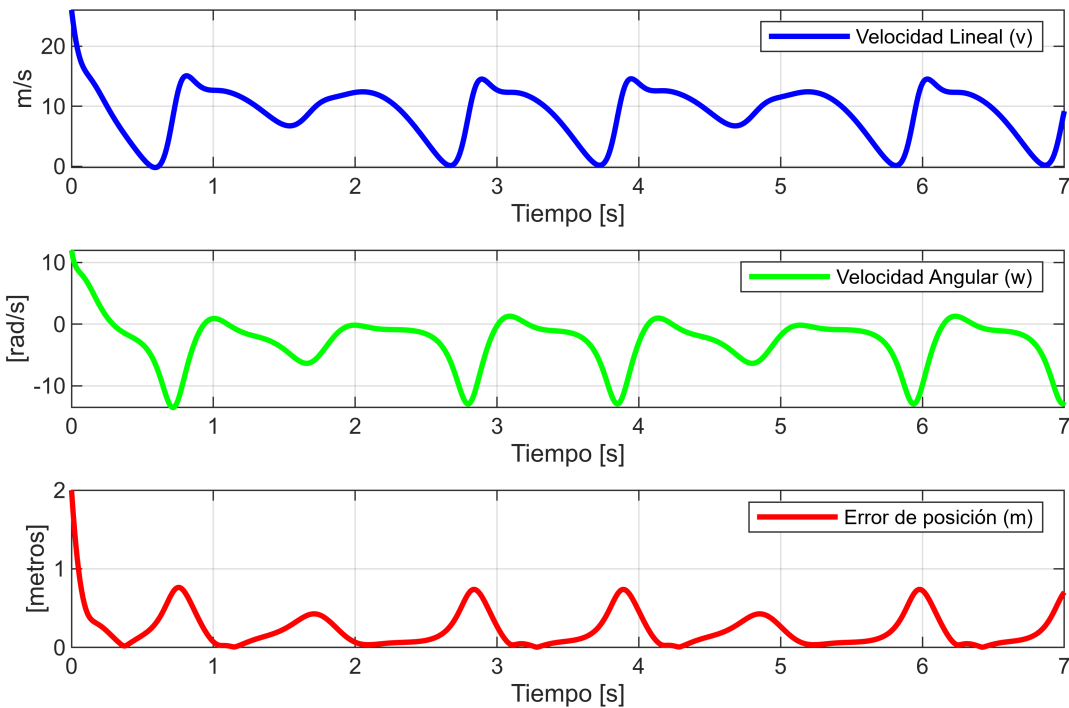
```
    pause(ts);
```

```
end
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
```

```
plot(t,Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('[metros]'),legend('Error de posición (m)');
```



```
%%
```

d) $x = \cos(t) + 1/2\cos(7t) + 1/3\sin(17t)$, $y = \sin(t) + 1/2\sin(7t) + 1/3\cos(17t)$

```
%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=10;           % Tiempo de simulación en segundos (s)
ts=0.01;         % Tiempo de muestreo en segundos (s)
t=0:ts:tf;       % Vector de tiempo
N= length(t);    % Muestras

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot
```

```

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1);          % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1);          % Posición del punto de control en el eje (Y) metros (m)

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=cos(t)+(1/2)*cos(7*t)+(1/3)*sin(17*t);
hyd=sin(t)+1/2*sin(7*t)+(1/3)*cos(17*t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[5 0;...
        0 5];

    %d)Velocidades deseadas
    hdp=[hxdp(k);hydp(k)];

    %e)Ley de Control: Agregamos las velocidades deseadas
    qpRef= pinv(J)*(hdp + K*he);

    v(k)= qpRef(1); %Velocidad lineal de entrada al robot
    w(k)= qpRef(2); %Velocidad angular de entrada al robot

```

```
%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
orientación
```

```
phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
xp1=v(k)*cos(phi(k));
```

```
yp1=v(k)*sin(phi(k));
```

```
%Aplico la integral a la velocidad lineal para obtener las cordenadas
```

```
% "x1" y "y1" de la posición
```

```
x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
```

```
y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)
```

```
% Posicion del robot con respecto al punto de control
```

```
hx(k+1)=x1(k+1);
```

```
hy(k+1)=y1(k+1);
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% a) Configuracion de escena
```

```
scene=figure; % Crear figura (Escena)
```

```
set(scene,'Color','white'); % Color del fondo de la escena
```

```
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
```

```
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
```

```
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
```

```
camlight('headlight'); % Luz para la escena
```

```
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
```

```
grid on; % Mostrar líneas de cuadrícula en los ejes
```

```
box on; % Mostrar contorno de ejes
```

```
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje
```

```
view([-0.1 90]); % Orientacion de la figura
```

```
axis([-2 2 -2 2 0 1]); % Ingresar limites minimos y maximos en los ejes x y z [minX
maxX minY maxY minZ maxZ]
```

```
% b) Graficar robots en la posicion inicial
```

```
scale = 4;
```

```
MobileRobot_5;
```

```
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;
```

```
% c) Graficar Trayectorias
```

```
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
```

```
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
```

```
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
```

```
% d) Bucle de simulacion de movimiento del robot
```

```
step=1; % pasos para simulacion
```

```
for k=1:step:N
```

```
    delete(H1);
```

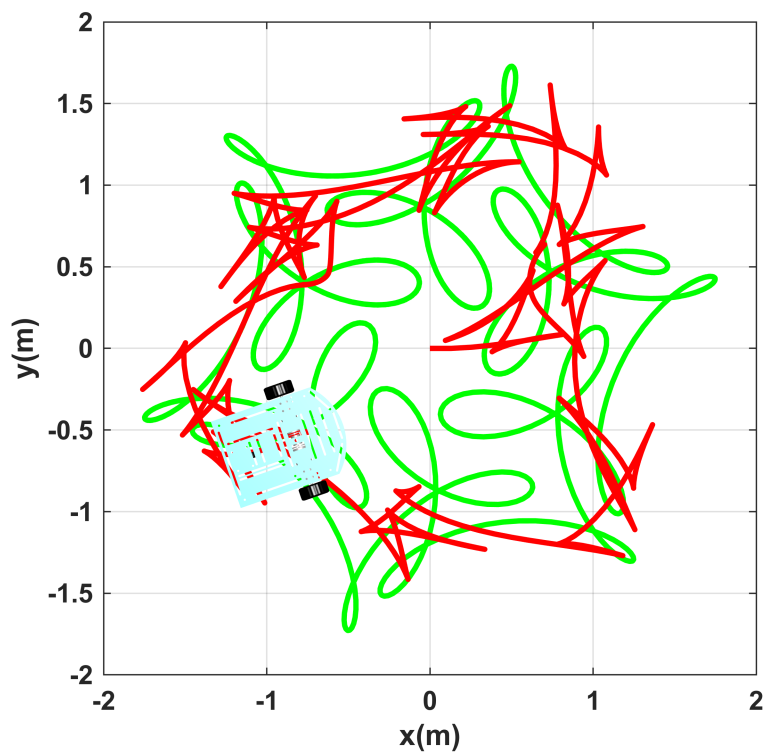
```
    delete(H2);
```

```
    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
```

```
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);
```

```
    pause(ts);
```

```
end
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
graph=figure; % Crear figura (Escena)
```

```
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
```

```
subplot(311)
```

```
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
```

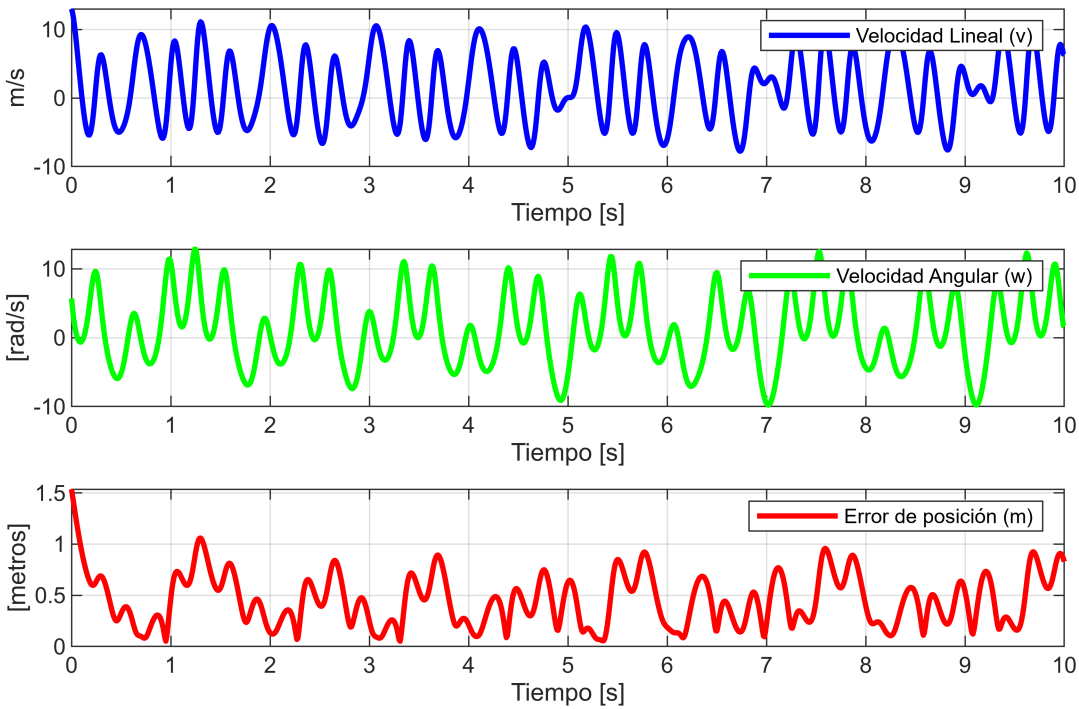
```
subplot(312)
```



```

plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[metros]'),legend('Error de posición (m)');

```



```
%%
```

e) $x = 17\cos(t) + 7\cos(17+7t)$, $y = 17\sin(t) - 7\sin(17+7t)$

```

%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=10;           % Tiempo de simulación en segundos (s)
ts=0.005;        % Tiempo de muestreo en segundos (s)
t=0:ts:tf;       % Vector de tiempo
N= length(t);    % Muestras

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x

```

```

y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=17*cos(t)+7*cos(17+7*t);
hyd=17*sin(t)-7*sin(17+7*t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[13 0;...
        0 13];

    %d)Velocidades deseadas
    hdp=[hxdp(k);hydp(k)];

    %e)Ley de Control:Agregamos las velocidades deseadas
    qpRef= pinv(J)*(hdp + K*he);

```

```

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
    orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

view([-0.1 90]); % Orientacion de la figura
axis([-25 25 -25 25 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;

```

```

MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

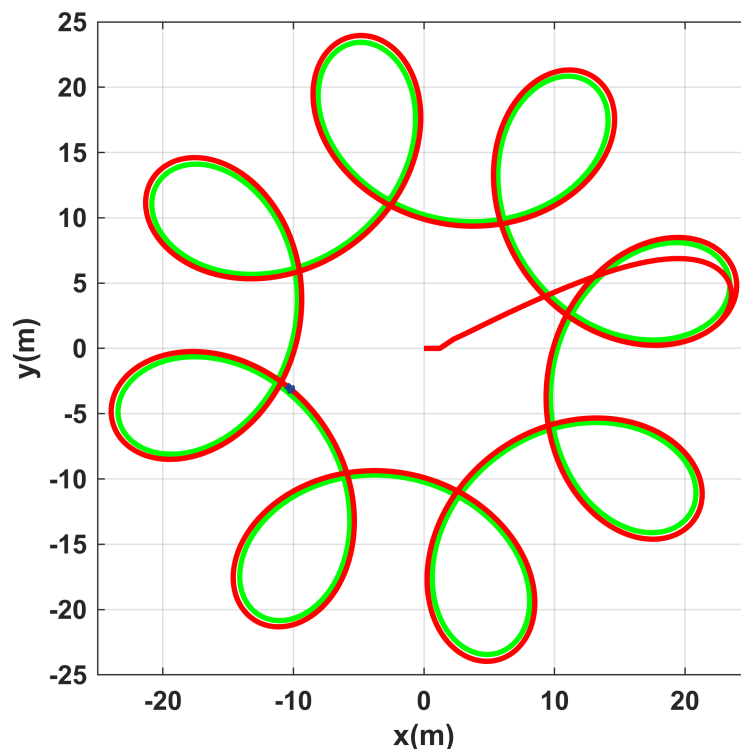
    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

```



```

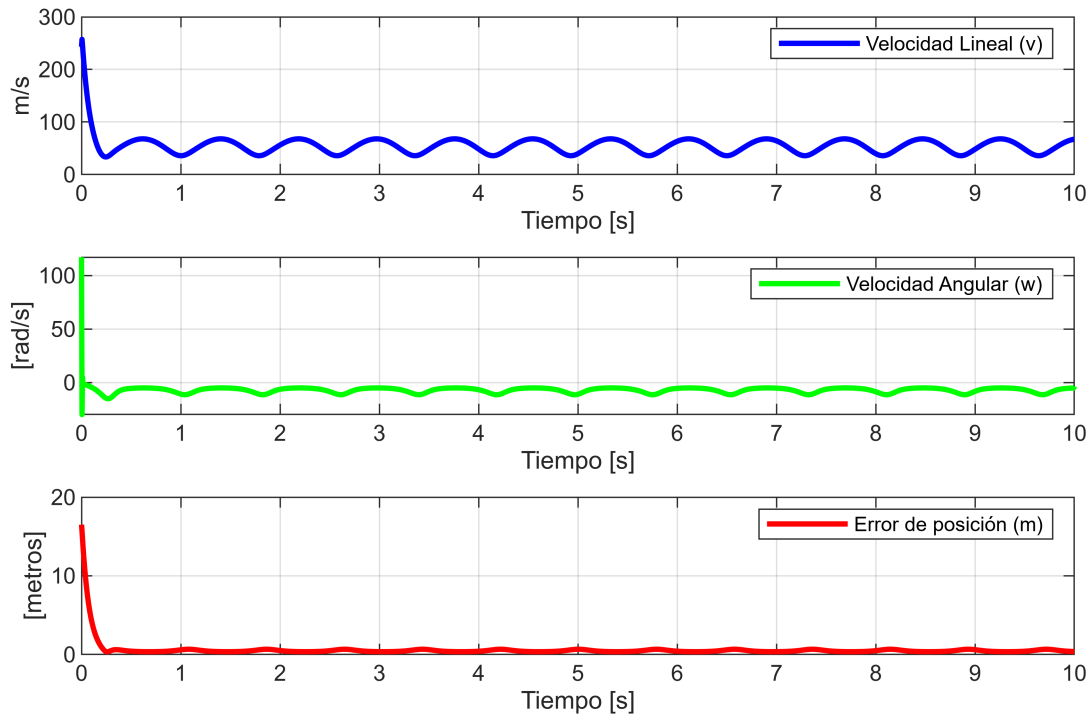
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)

```

```

plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/
s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('rad/
s'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('metros'),legend('Error de posición (m)');

```



f) $x = 2\cos(t)$, $y = 2\sin(t)$

```

%%
%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=40;           % Tiempo de simulación en segundos (s)
ts=0.02;         % Tiempo de muestreo en segundos (s)
t=0:ts:tf;       % Vector de tiempo
N= length(t);    % Muestras

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=2*cos(t);
hyd=2*sin(t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[13 0;...
        0 13];

    %d)Velocidades deseadas
    hdp=[hxdp(k);hydp(k)];

```

```

%e)Ley de Control:Agregamos las velocidades deseadas
qpRef= pinv(J)*(hdp + K*he);

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

view([-0.1 90]); % Orientacion de la figura
axis([-10 10 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

```

```

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

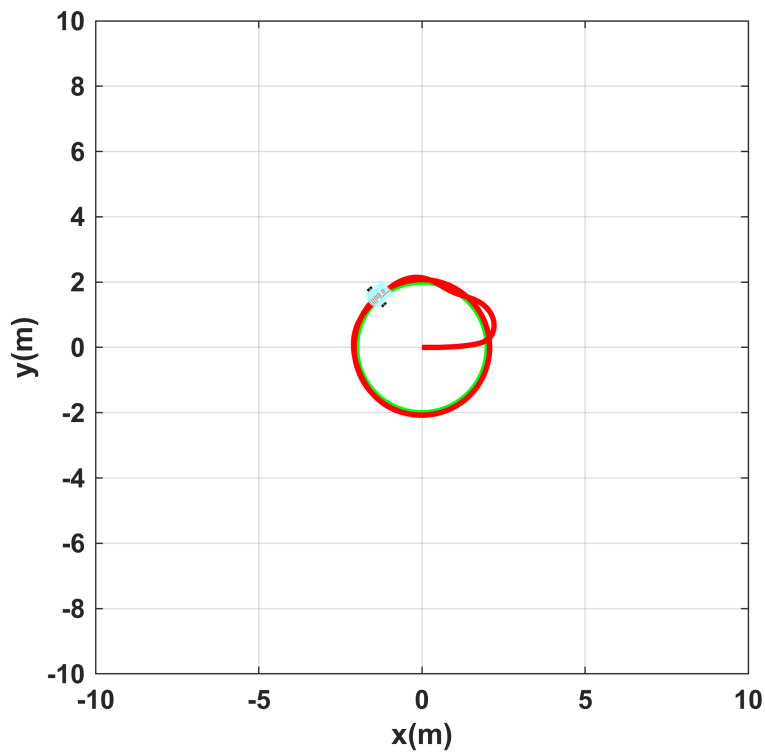
    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)

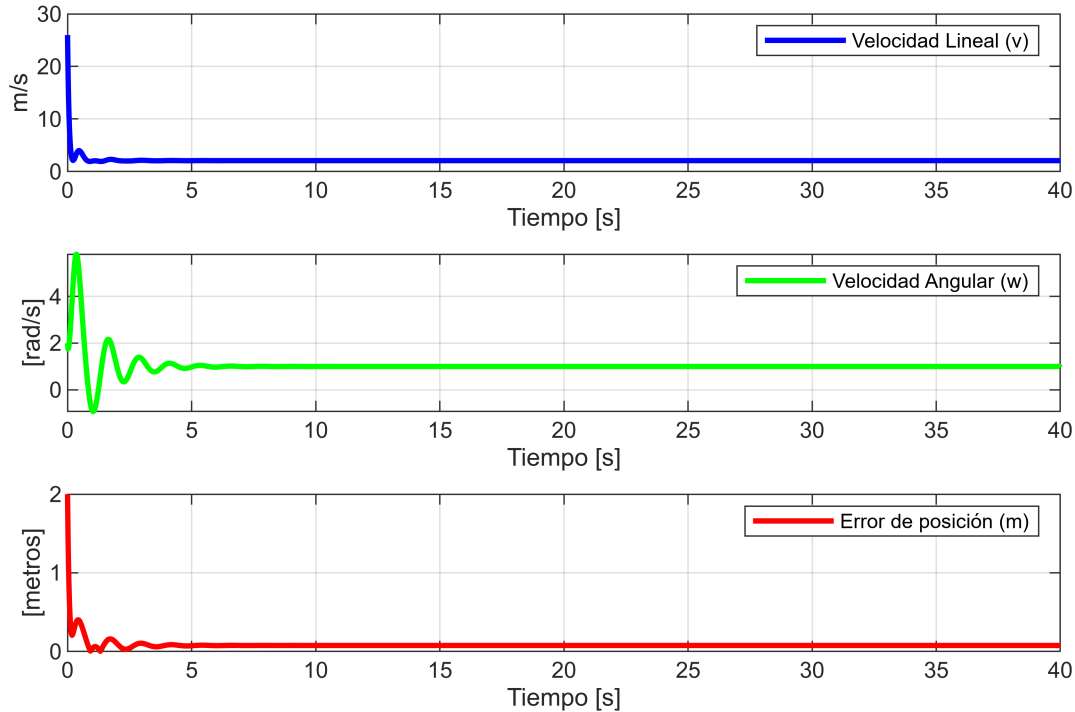
```



```

set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[metros]'),legend('Error de posición (m)');

```



```
%%
```

g) $x = 5t - 4\sin(t)$, $y = 5t - 4\cos(t)$

```

%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=40;          % Tiempo de simulación en segundos (s)
ts=0.05;        % Tiempo de muestreo en segundos (s)
t=0:ts:tf;      % Vector de tiempo
N= length(t);   % Muestras

```

```

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=5*t-4*sin(t);
hyd=5*t-4*cos(t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[13 0;...
        0 13];

```

```

%d)Velocidades deseadas
hdp=[hxdp(k);hydp(k)];

%e)Ley de Control:Agregamos las velocidades deseadas
qpRef= pinv(J)*(hdp + K*he);

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
    orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

view([-0.1 90]); % Orientacion de la figura

```

```

axis([-10 75 -10 75 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

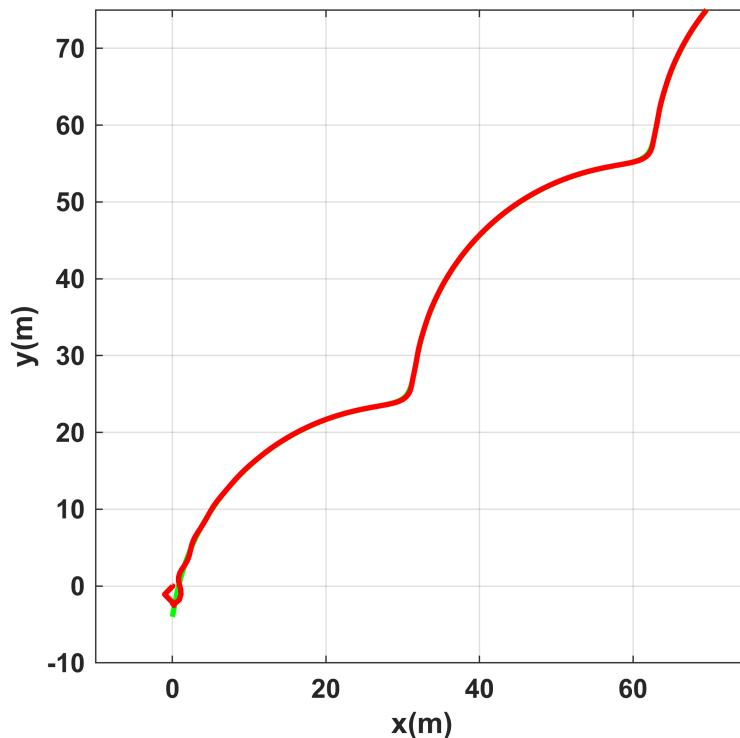
    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

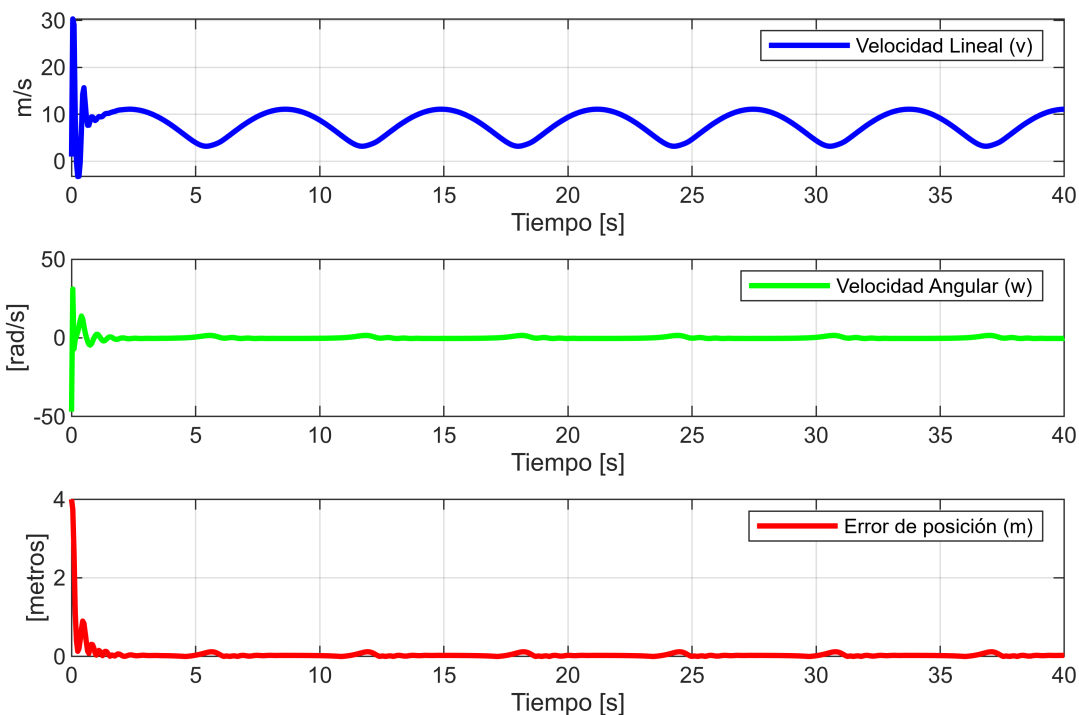
```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[metros]'),legend('Error de posición (m)');

```



h) $x = 4\cos(t) + \cos(4t)$, $y = 4\sin(t) - \sin(4t)$

```

%%
%Limpieza de pantalla
clear all
close all
clc

%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=10; % Tiempo de simulación en segundos (s)
ts=0.02; % Tiempo de muestreo en segundos (s)
t=0:ts:tf; % Vector de tiempo
N= length(t); % Muestras

```

```

%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot

%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)

%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Ecuaciones paramétricas de la trayectoria deseada
%Figura 1
hxd=4*cos(t)+cos(4*t);
hyd=4*sin(t)-sin(4*t);

%Velocidades de la trayectoria deseada
% Velocidades de la trayectoria deseada (derivadas respecto al tiempo)
hxdp = diff(hxd)/ts;
hydp = diff(hyd)/ts;

% Ajustamos tamaño para que coincidan con t
hxdp(N) = hxdp(N-1); % Último valor repetido
hydp(N) = hydp(N-1); % Último valor repetido

for k=1:N

    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);

    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

    %b)Matriz Jacobiana
    J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
        sin(phi(k)) cos(phi(k))];

    %c)Matriz de Ganancias
    K=[13 0;...

```

```

    0 13];

%d)Velocidades deseadas
hdp=[hxdp(k);hydp(k)];

%e)Ley de Control:Agregamos las velocidades deseadas
qpRef= pinv(J)*(hdp + K*he);

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
    orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

```

```

view([-0.1 90]); % Orientacion de la figura
axis([-10 10 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

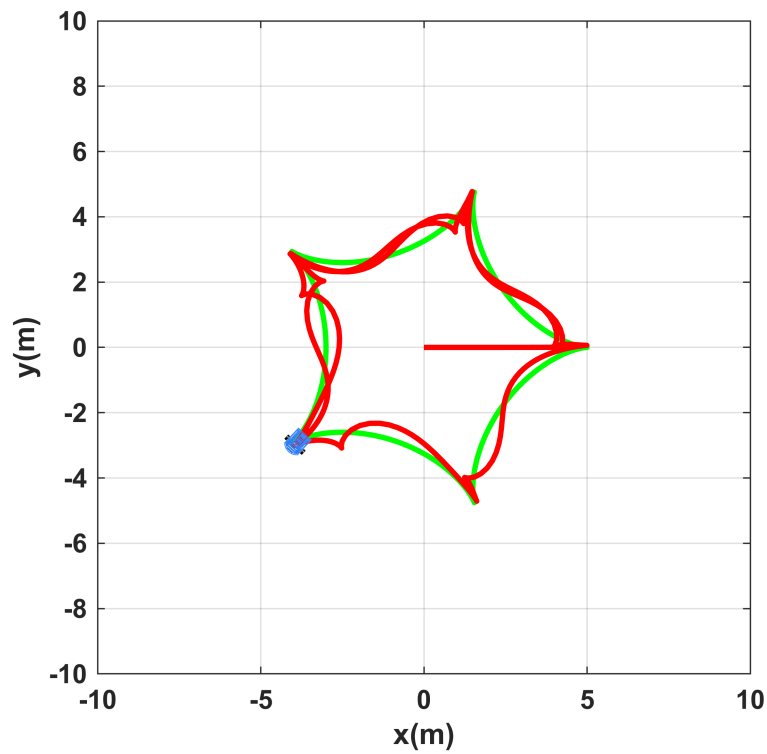
    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/s]'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[metros]'),legend('Error de posición (m)');

```

